

Prosiding  
**ANNUAL RESEARCH SEMINAR 2016**  
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

# Klasifikasi Malware Trojan Ransomware Dengan Algoritma Support Vector Machine (SVM)

Erick Lamdompak S  
Sistem Komputer, Fakultas Ilmu Komputer  
Universitas Sriwijaya  
Palembang, Indonesia  
[ericklamdompak@gmail.com](mailto:ericklamdompak@gmail.com)

**Abstrak**— Identifikasi dari file malware itu sendiri merupakan proses yang kompleks, yang melibatkan ekstraksi ciri khas dari satu sampel dataset malware. Dataset normal file dan dataset malware Trojan ransomware diperiksa menggunakan pola opcode n-gram yang diekstrak dari file setelah pembongkaran. Opcode pola n-gram digunakan sebagai fitur untuk proses klasifikasi. Penelitian yang dilakukan dalam makalah ini adalah melakukan training untuk normal file sebagai pembandingan dengan file Trojan ransomware yang di deteksi menggunakan fitur ekstraksi N-grams. Hasil dari analisis sementara proses training dan deteksi malware harus di dukung dengan metode klasifikasi yang akurat dan efisien. Hasil dari penelitian ini menunjukkan bahwa Support vector Machine (SVM) adalah metode yang kuat dan akurat untuk penyelesaian klasifikasi malware Trojan Ransomware dan normal file.

**Kata kunci**— Trojan Ransomware, klasifikasi malware, N-grams, Support Vector Machine.

## I. PENDAHULUAN

Trojan Ransomware merupakan salah satu jenis malware yang paling berbahaya dan paling dibenci oleh para pengguna komputer, dikarenakan Trojan Ransomware ini menggunakan kode berbahaya untuk mengganggu sistem sebelum pengguna menyadarinya, dapat mengenkripsi file penting seseorang dan meminta nya untuk membayar ke pada si penyerang agar file itu bisa di gunakan lagi oleh penggunanya hingga merusak seluruh hard drive pengguna tersebut[1]. Terdapat sepuluh malware yang paling terpopuler di tahun 2015 adalah Trojan Ransomware, Exploit kits, Banking Trojans, worms, PoS (Point-of-Sale) Malware, Social Engineering Attacks, Fake Tech Support Services, Rogue Antivirus Software, Potentially Unwanted Programs, dan Adware[5].

Teknik analisis dalam pendeteksian malware umumnya diklasifikasikan menjadi pendekatan dinamis dan statis. Dalam analisis dinamis dikenal sebagai analisis perilaku sedangkan dalam analisis statis, informasi mengenai program atau perilaku yang diharapkan dalam kode biner [2]. Support Vector Machine adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi dalam analisis dinamis maupun statis[3]. SVM memiliki prinsip dasar linier classifier yaitu kasus klasifikasi yang secara linier dapat dipisahkan, namun SVM telah dikembangkan agar dapat bekerja pada problem non-linier dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi[4]

Tujuan dari penelitian ini adalah menggunakan Support Vector Machine untuk klasifikasi malware dan normal file sebagai pembandingan yang di dapat dari dataset malware dan dataset file normal. Dilakukannya pengklasifikasian setiap sampel data kedalam salah satu dari dua kategori yaitu kelas positif yang dilambangkan dengan 1 sebagai data normal dan kelas negatif dilambangkan dengan -1 sebagai data berbahaya (malware) yang dipisahkan dengan garis hyperlane. Malware dapat dideteksi dengan menggunakan fitur N-grams pola sequensial yang digunakan setelah melakukan pembongkaran file malware dalam dataset.

## II. GAMBARAN KERJA

Pada penelitian [6] membahas bagaimana banyak nya data yang sangat beragam (heterogeneous data) yang dianggap sebagai data ancaman dari internet. Penelitian ini menggunakan Support Vector Machine untuk klasifikasi heterogeneous data atau data yang beragam dengan mengklasifikasikan setiap sampel data ke dalam salah satu dari dua kategori: kelas positif dilambangkan dengan "1" (data normal) dan kelas negatif dilambangkan dengan "-1" (data berbahaya).

Prosiding  
**ANNUAL RESEARCH SEMINAR 2016**  
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

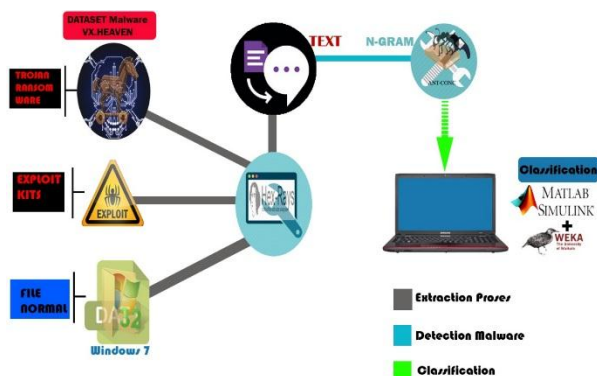
<http://ars.ilkom.unsri.ac.id>

Pada penelitian lain [7] membahas bagaimana SVM digunakan dalam melakukan identifikasi *malware* yang dapat di deteksi dengan menggunakan *Support Vector Machine* (SVM) dan kode operasional yang terdapat pada *malware* (opcodes), yang menunjukan bahwa (ja, adc, sub, inc, add, mov and rep) adalah indikator kuat *malware* yang memberikan dasar untuk peningkatan teknik mendeteksi *malware* di beberapa negara. Dan pada penelitian ini juga membahas bagaimana tahap identifikasi penyaringan menggunakan fitur penyaringan *vector eigen* atau (*eigen vector* 'prefilter').

Pada penelitian lainnya [8] membahas bagaimana proses pencegahan identifikasi karakteristik unik dari program *malware* seperti enkripsi dan pola kebingungan. Pada penelitian ini menggunakan N-grams sebagai fitur sequential setelah proses dataset *malware* di bongkar untuk mendapatkan pola dari keluarga *malware*. Teknik klasifikasi yang di gunakan ada 3 yaitu: Decision tree (C4.5), ANN dan SVM untuk menentukan metode manakah yang memiliki tingkat akurasi yang paling tinggi.

### III. METODOLOGI

Penelitian ini berfokus pada klasifikasi *malware* dengan beberapa fitur pilihan yang digunakan untuk mewakili setiap sampel *malware* dalam pengklasifikasian. Perancangan sistem penelitian ini terdapat pada gambar 1 berikut:



Gambar 1. Perancangan sistem penelitian

Penelitian ini hanya menggunakan 1 perangkat fisik yang difungsikan untuk pengolahan data menggunakan matlab+weka sebagai media klasifikasi dan IDA pro[18] dan

N-grams menggunakan knfgram[19] dan Ant conc [20] sebagai diassemble file dan ekstraksi.

#### A. Diassemble File

*Interactive Disassembler (IDA)* Pro adalah diassembler untuk software komputer yang menghasilkan kode operasional bahasa *assembly* dari kode mesin *executable* atau biasanya kita sebut .exe [28].

Berikut hasil dari *Diassemble file* pada dataset *malware Trojan Ransomware* dan normal file:

##### 1. Normal set Representation

.rdata:0040B940	20 54 68 69 73 20 61 70	70 6C 69 63 61 74 69 6F
.rdata:0040B950	6E 20 63 61 6E 6E 6F 74	20 72 75 6E 20 75 73 69
.rdata:0040B960	6E 67 20 74 68 65 20 61	63 74 69 76 65 20 76 65
.rdata:0040B970	72 73 69 6F 6E 20 6F 66	20 74 68 65 20 4D 69 63
.rdata:0040B980	72 6F 73 6F 66 74 20 2E	4E 45 54 20 52 75 6E 74
.rdata:0040B990	69 6D 65 0A 50 6C 65 61	73 65 20 63 6F 6E 74 61

Gambar 2. Tampilan Hexa normal file

```
mov     edi, edi
push    ebp
mov     ebp, esp
sub     esp, 14h
push    esi
push    edi
xor     edi, edi
mov     [ebp+dwRegister], edi
mov     [ebp+var_14], edi
mov     [ebp+dwMilliseconds], 0BB8h
mov     esi, offset unk_1014458
push    esi
call    sub_10030A7
```

Gambar 3. Tampilan *assembly* normal file

Dari gambar 3 dapat dijelaskan bahwa hasil dari training file normal menghasilkan kode operasional yang sangat berbeda dengan hasil dari *malware Trojan Ransomware* yang telah dibongkar. Perbedaannya sangat signifikan dilihat dari kode operasional yang berbeda pada hasil *assembly* di file normal.

##### 2. Malware Representation

HEADER:00400040	BA 10 00 0E 1F B4 09 CD	21 B8 01 4C CD 21 90 90
HEADER:00400050	54 68 69 73 20 70 72 6F	67 72 61 6D 20 6D 75 73

*Prosiding*  
**ANNUAL RESEARCH SEMINAR 2016**  
 6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

http://ars.ilkom.unsri.ac.id

```

HEADER:00400060 74 20 62 65 20 72 75 6E 20 75 6E
        64 65 72 20 57
HEADER:00400070 69 6E 33 32 0D 0A 24 37 00 00 00
        00 00 00 00 00
  
```

Gambar 4. Tampilan hexa malware Trojan ransomware

```

push    ebx
mov     eax, [ebp+nCmdShow]
push    esi
mov     dword_40C04C, eax
push    edi
xor     edi, edi
mov     [ebp+var_2], di
cmp     [ebp+arg_4], edi
jnz     short loc_401090
mov     [ebp+var_54.cbClsExtra], edi
mov     esi, [ebp+hInstance]
push    offset IconName ; "SC"
push    esi ; hInstance
mov     [ebp+var_54.hInstance], esi
mov     [ebp+var_54.cbSize], 30h
mov     [ebp+var_54.lpfnWndProc], offset sub_40175F
mov     [ebp+var_54.cbWndExtra], 1Eh
mov     ecx, 0A6Bh
mov     [ebp+var_54.lpszClassName], offset AppName ;
"SciCalc"
lea     eax, [ebp+var_54]
push    eax ; WNDCLASSEX *
call    ds:RegisterClassExA
test    ax, ax
jnz     short loc_401093
jmp     loc_401372
  
```

Gambar 5. Tampilan Assembly malware Trojan

Dari gambar 5 diatas dapat dijelaskan bagaimana perbedaan konten dalam malware dan normal file yang hanya di jumpai dalam file *malware* saja, sebagai berikut:

1. Pada gambar di atas kode xor yang di gunakan adalah (xor edi, edi) yang merupakan bagian dari proses *decrypt virus malware Trojan* yang sudah di deklarasikan pada bagian awal *script*.
2. kode operasional pada *Trojan Ransomware* di gambar 5 yaitu: cmp [ebp+arg\_4], edi, kode operasional tersebut membandingkan kode ebp dengan edi dan akan memverifikasi kode manakah yang akan diteruskan untuk ke sesi berikutnya pada proses malware dijalankan.
3. Kode mov ecx, 0A6Bh kode operasional ini merupakan panjang dari bagian badan atau *body script* enkripsi dan fungsi dari kode tersebut merupakan bagian dari proses enkripsi file yang akan diserang oleh malware Trojan Ransomware.
4. Pada kode operasional Trojan ransomware di gambar 5 letak kode jnz dan jmp setelah kode operasional test

ax,ax. jnz short loc\_401093, jmp loc\_401372 kode ini dapat diartikan bahwa jika lokasi tujuan tidak sesuai, kode berikut nya akan pindah ke lokasi yang akan di set sebagai breakpoint dan akan meneruskan ke kode berikutnya.

## B. Feature Extraction

### 1. N-gram Extraction

Setelah file di ekstrak ke dalam sebuah string panjang *hexadecimal* dan *assembly* menggunakan IDA pro, kemudian string diolah menjadi satu set tumpang tindih dengan N-grams. Manfaat dari menggunakan n-gram adalah bahwa hal itu dapat menangkap frekuensi kata-kata yang memiliki panjang n-gram [21]. Banyak penelitian ke dalam fitur n-gram telah menyarankan 4-gram untuk menjadi yang terbaik[8].

Dalam penelitian ini digunakan fitur n-gram dengan *range*  $n = 1$ ,  $n = 2$ ,  $n = 3$  dan  $n = 4$ . Dan hasil dari fitur N-gram ini dapat di lihat pada tabel berikut:

No	Freq	N-grams =1	No	Freq	N-grams = 2
1	24	Ebp	1	15	Ebp var
2	19	Mov	2	11	Mov ebp
3	17	Var	3	5	Eax ebp
4	15	eax	4	4	Call sub
5	11	push	5	4	Dword ptr
6	8	ptr	6	4	Mov eax
7	8	sub	7	3	Eax mov
8	7	edi	8	3	Edi mov
9	6	esp	9	3	Push eax
10	6	pop	10	3	Var push

Tabel 1. Hasil N-gram Extraction

Setelah file berhasil dibongkar dan di ekstraksi dan menghasilkan kode operasional (*opcodes*), N-grams digunakan untuk mengurutkan hasil ekstraksi kode operasional yang muncul dalam file Trojan Ransomware dan File Normal dengan mengabaikan lokasi, memory, dan register. Contoh:

*Prosiding*  
**ANNUAL RESEARCH SEMINAR 2016**  
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

`mov dword 40C04C, eax` akan di normalisasikan dengan mengurutkan kode operasional `mov` saja berdasarkan frekuensi banyaknya `mov` muncul di file *Trojan Ransomware* dan File normal, tetapi pengabaian lokasi, memory dan register ini hanya berlaku jika N-gram menggunakan range  $n=1$ .

### C. Classification

Klasifikasi adalah proses dimana klasifier (user yang melakukan pengklasifikasian) belajar dari data sampel yang telah di beri label, data tersebut digunakan untuk pengujian sampel data. Setiap contoh malware di dalam dataset nilai satu target dan beberapa atribut, proses keseluruhan dari klasifikasi ini melibatkan penggunaan metode mesin belajar (*machine learning*) untuk hasil klasifikasi file yang tidak diketahui, apakah jinak atau berbahaya.

Metode atau teknik dalam pengklasifikasian adalah *Support Vector Machine*, SVM merupakan teknik pembelajaran yang cocok untuk memecahkan masalah klasifikasi dengan ruang fitur dimensi tinggi, SVM adalah teknik yang digunakan untuk data klasifikasi dan dikategorikan sebagai metode kernel.

#### 1. Support Vector Machine

Dalam penelitian ini menggunakan implementasi LIBSVM dengan fungsi kernel polinomial untuk melatih SVM walaupun tetap menggunakan teknik dasar digunakan untuk klasifikasi biner. selama fase pelatihan, algoritma memperoleh pengetahuan tentang kelas dengan memeriksa training set yang menggambarkan dataset tersebut. Selama fase pengujian, mekanisme klasifikasi meneliti data set dan sekumpulan anggota nya untuk kelas yang tersedia. Target dari algoritma ini adalah estimasi batas antara kelas.

Dataset *Trojan ransomware* terdiri dari 149 *opcodes* yang berbeda, dan masing-masing memiliki karakteristik distribusi yang unik, karena itu SVM adalah pilihan yang tepat untuk melakukan klasifikasi

## IV. DATASET

### A. Dataset Malware VX Heavens Virus Collection

Dataset yang digunakan dalam penelitian ini adalah *VX Heavens Virus Collection database* [27] yang memiliki bermacam-macam jenis *malware* dan *virus* yang telah di kelompokkan dalam tabel berikut:

Malware Class	Malware Type
Trojan	Banker, Clicker, DDoS, Downloader, Dropper, GameThief, IM, Mailfinder, Notifier, PSW, Proxy, Ransomware, SMS, Spy, Win32.
Exploit	Exploit.DOS, Exploit.HTA.Behind, IRC, HTML, Exploit.IIS.Beavuh, JS,Java, Linux, MSPPoint, MSWord, Exploit.Mac.Small, Exploit.Multi.Qtp, Exploit.MySQL.Privilege, PHP, Perl, Python, SWF, Shell, SunOS, Unix, VBS, Win16, Win32
Flooder	Flooder. DOS.Code, IRC, Java, Linux, Flooder.PHP.MailSpam, Flooder.Unix.MailSpam, Win16,Win32, Email-Flooder.BAT.Prob, Email-Flooder.DOS.Bombing, Email-Flooder.JS.Viv, Perl, IM-Flooder, SMS-Flooder
Constructor	BAT, MSIL, HTML, DoS, Perl, MSWord, Script, VBS, Win16, Win32
Backdoor	ASP, BAT, IRC, Java, Linux, MSIL, PHP, Perl, Python, SunOS, Unix, VBS, Win16, Win32
Virus	ABAP, ALS, ALS, Acad, AmiPro, BAS, BAT, Boot-DOS, Boot, CSC, DOS, DOS32, Ferite, HTML, JS, Java, Linux, Lua, MSAccess, MSEXcel, MSH, MSIL, MSOffice, MSPPoint, MSPProject, MSVisio, MSWord, Mac, Matlab, Menuet, Multi, OS2, PHP, Perl, Python, QNX, Ruby, SGold, Script, StarOffice, TSQL, Unix, VBS, WBS, Win16, Win32, Win64, Win9x, WinHLP, WinINF, WinREG
Worm	FreeBSD, Email-Worm, IM-Worm, IRC-Worm, Net-Worm, P2P-Worm

Tabel 2. Malware Class pada VX Heaven Virus Collection Dataset

### B. Dataset Opcodes (Operational Codes)

Kode operasional (*Opcodes*) adalah instruksi bahasa mesin yang melakukan operasi CPU pada operan seperti aritmatika, manipulasi memori / data, operasi logis dan Program flow control [7].

Opcode Class	Opcodes Type
Arithmetic operations	<i>add, adc</i> ( <i>add</i> dengan tambahan

Prosiding  
**ANNUAL RESEARCH SEMINAR 2016**  
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

http://ars.ilkom.unsri.ac.id

	<i>flag), inc, sub, cmp, dec, div, idiv, imul</i>
<i>Memory manipulation</i>	<i>lea (load effective address), mov, pop, push, (memasukkan nilai dari Operand ke dalam Stack)</i>
<i>Logical operations</i>	<i>xor (exclusive or), and, or</i>
<i>Program flow control</i>	<i>call (jump to a function), ret (re-turn from function) cmp (compare data), ja, je, jmp (jump if a condition is met)</i>

Tabel 3. Opcodes (kode Operasional di dalam malware C. Registers

Sebelum mengetahui tentang kode operasional *assembly* yang paling dasar, terlebih dahulu harus akrab dengan register IA-32 bit. Pada umumnya dalam IA-32 bit ada 8 register umum[29] yaitu: EAX, EBX, ECX, EDX, ESI, EDI, EBP, dan ESP. Diluar register umum tersebut, arsitektur juga mendukung *floating point register* dan *register* lainnya diluar 8 register tersebut yang melayani tingkat kebutuhan dari sistem yang digunakan.

## V. HASIL SEMENTARA

### A. Trace Dataset

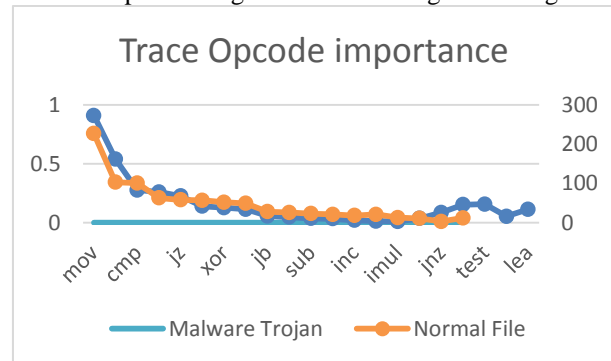
Berikut hasil trace pada file Trojan Ransomware dalam file zakk.a dan normal file mcbuilder.exe:

Gambar 6. Hasil trace Trojan Ransomware pada thread

Thread	Address	Instruction	Result
0000128C	0000128C	mov [esp+4], eax	ST0=0.0ST1=0.0ST2=0.0ST3=0.0ST4=0.0ST5=0.0...
0000128C	0000128C	mov [esp+8], ebx	ST0=0.0ST1=0.0ST2=0.0ST3=0.0ST4=0.0ST5=0.0...
0000128C	0000128C	mov [esp+12], ecx	EFL=202
0000128C	0000128C	mov [esp+16], edx	
0000128C	0000128C	push esp	ESP=1CFFEC
0000128C	0000128C	mov ebp, esp	EBP=1CFFEC
0000128C	0000128C	push ecx	ESP=1CFFEB
0000128C	0000128C	push ecx	ESP=1CFFEA
0000128C	0000128C	lea eax, [ebp+8]	EAX=1CFFEA
0000128C	0000128C	push eax	ESP=1CFFE9
0000128C	0000128C	call near ptr ntdll!_RtlInitializeExceptionChain	ESP=1CFFD0
0000128C	0000128C	call near ptr ntdll!_RtlInitializeExceptionChain	
0000128C	0000128C	mov edi, edi	ESP=1CFFD0
0000128C	0000128C	push ebp	ESP=1CFFD0
0000128C	0000128C	mov ebp, esp	PF=1 2F=1
0000128C	0000128C	cmp dword_77D7C0B0, 1	EBP=1CFFD0 ESP=1CFFD0
0000128C	0000128C	jnz near ptr unk_77D7C0B0	ESP=1CFFD0
0000128C	0000128C	pop ebp	ESP=1CFFD0
0000128C	0000128C	ret 4	ESP=1CFFD0
0000128C	0000128C	with 4	ESP=1CFFD0
0000128C	0000128C	push dword ptr [ebp+0Ch]	ESP=1CFFD0
0000128C	0000128C	push dword ptr [ebp+8]	ESP=1CFFD0
0000128C	0000128C	call sub_77C7B3CE	ESP=1CFFD0

Thread	Address	Instruction	Result
0000128C	0000128C	mov [esp+4], eax	ST0=0.0ST1=0.0ST2=0.0ST3=0.0ST4=0.0ST5=0.0...
0000128C	0000128C	mov [esp+8], ebx	ST0=0.0ST1=0.0ST2=0.0ST3=0.0ST4=0.0ST5=0.0...
0000128C	0000128C	mov [esp+12], ecx	EFL=202
0000128C	0000128C	mov [esp+16], edx	
0000128C	0000128C	push esp	ESP=1CFFEC
0000128C	0000128C	mov ebp, esp	EBP=1CFFEC
0000128C	0000128C	push ecx	ESP=1CFFEB
0000128C	0000128C	push ecx	ESP=1CFFEA
0000128C	0000128C	lea eax, [ebp+8]	EAX=1CFFEA
0000128C	0000128C	push eax	ESP=1CFFE9
0000128C	0000128C	call near ptr ntdll!_RtlInitializeExceptionChain	ESP=1CFFD0
0000128C	0000128C	call near ptr ntdll!_RtlInitializeExceptionChain	
0000128C	0000128C	mov edi, edi	ESP=1CFFD0
0000128C	0000128C	push ebp	ESP=1CFFD0
0000128C	0000128C	mov ebp, esp	PF=1 2F=1
0000128C	0000128C	cmp dword_77D7C0B0, 1	EBP=1CFFD0 ESP=1CFFD0
0000128C	0000128C	jnz near ptr unk_77D7C0B0	ESP=1CFFD0
0000128C	0000128C	pop ebp	ESP=1CFFD0
0000128C	0000128C	ret 4	ESP=1CFFD0
0000128C	0000128C	with 4	ESP=1CFFD0
0000128C	0000128C	push dword ptr [ebp+0Ch]	ESP=1CFFD0
0000128C	0000128C	push dword ptr [ebp+8]	ESP=1CFFD0
0000128C	0000128C	call sub_77C7B3CE	ESP=1CFFD0

Gambar 7. Hasil trace pada normal file pada 2 thread  
Hasil dari trace dataset malware dan normal file dilakukan perbandingan dalam bentuk grafik sebagai berikut:



Gambar 7. Perbandingan kode operasional pada trace file

Grafik diatas merupakan hasil dari kode operasional (opcodes) yang di peroleh dari proses trace yang dilakukan pada normal file dan Trojan malware. Dapat kita liat letak perbedaan yang signifikan dari jumlah kode operasional yang berada di file Trojan dan normal. Mov, Push dan cmp merupakan kode operasional yang paling sering muncul di antara kedua file, tetapi yang paling membedakan adalah pada kode operasioanl test dan lea yang tidak tedapat pada file normal.

### B. Pengujian Data menggunakan Support Vector Machine

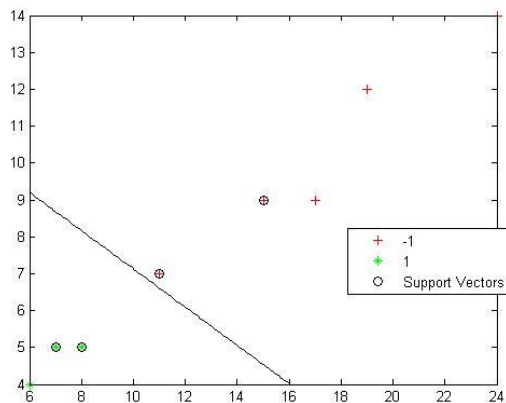
Pengujian data yang diperoleh dari hasil N-gram Extraction dan terlihat perbedaan dari segi frekuensi nya yang dapat di tampilkan pada hasil klasifikasi berikut:



*Prosiding*  
**ANNUAL RESEARCH SEMINAR 2016**  
*6 Desember 2016, Vol 2 No. 1*

ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>



Gambar 8. Hasil klasifikasi dengan SVM

Dari hasil klasifikasi di atas terlihat bahwa malware dan normal file terpisah berdasarkan garis hyperlane yang berada di tengah, dan malware dilambangkan dengan -1 yang berwarna merah sedangkan normal file dengan 1 yang berwarna hijau.

#### VI. KESIMPULAN SEMENTARA

Adapun kesimpulan sementara yang didapatkan dari penelitian ini adalah sebagai berikut:

1. N-grams dapat mendeteksi malware yang di ekstrak berdasarkan kode operasional yang sering muncul.
2. Algoritma Support Vector Machine menghasilkan klasifikasi antara Malware dengan Normal file yang di pisahkan berdasarkan garis Hyperlane.

Malware dan normal file di klasifikasikan berdasarkan kelas -1 untuk malware dan 1 untuk normal file.

#### REFERENSI

- [1] X. Luo and Q. Liao, "Ransomware : a new cyber hijacking threat to enterprises," in *Handbook of Research on Information Security and Assurance*, IGI Global, 2009.
- [2] Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. (2008). Learning and classification of malware behavior. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5137 LNCS, 108–125. [https://doi.org/10.1007/978-3-540-70542-0\\_6](https://doi.org/10.1007/978-3-540-70542-0_6)
- [3] Octaviani, P. A., Yuciana Wilandari, & Ispriyanti, D. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. *Jurnal Gaussian - Issn: 2339-2541*, 3(8), 811–820. Retrieved from <http://ejournal-s1.undip.ac.id/index.php/gaussian>
- [4] Noorniawati, V. Y. (2007). Metode Support Vector Machine untuk Klasifikasi pada Sistem Temu Kembali Citra, 64103034–64103034.
- [5] CFOC.ORG. (2016). Top 10 Malware Attacks Retrieved from <http://cfoc.org/top-10-malware-attacks-of-2015/>
- [6] Kruczkowski, M., & Szykiewicz, E. N. (2014). Support Vector Machine for Malware Analysis and Classification. *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 415–420. <http://doi.org/10.1109/WI-IAT.2014.127>
- [7] O’Kane, P., Sezer, S., McLaughlin, K., & Im, E. G. (2013). SVM Training phase reduction using dataset feature filtering for malware detection. *IEEE Transactions on Information Forensics and Security*, 8(3), 500–509. <https://doi.org/10.1109/TIFS.2013.2242890>
- [8] Liangboonprakong, C., & Sornil, O. (2013). Classification of malware families based on N-grams sequential pattern features. *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, 777–782. <http://doi.org/10.1109/ICIEA.2013.6566472>
- [18] IDA Support: Download Center. (n.d.). Retrieved from <https://www.hex-rays.com/products/ida/support/download.shtml>
- [19] Kfnggram. (n.d.). Retrieved from <http://www.kwicfinder.com/kfNgram/kfNgramHelp.html>
- [20] Ant Conc. (n.d.). Retrieved from <http://www.laurenceanthony.net/software/antconc/>
- [21] Jain, S., & Meena, Y. K. (2011). Computer Networks and Intelligent Computing: 5th International Conference on Information Processing, ICIP 2011, Bangalore, India, August 5-7, 2011. *Proceedings. Computer Networks and Intelligent Computing*, 51–59. [http://doi.org/10.1007/978-3-642-22786-8\\_6](http://doi.org/10.1007/978-3-642-22786-8_6)
- [27] Heaven, V. (n.d.). Computer Virus Collection. Retrieved from <http://vxheaven.org/vl.php>
- [29] M. Christodorescu, S. Jha, S. Seshia, D. Song, and R. Bryant. Semantics-aware malware detection. In *Security and Privacy, 2005 IEEE Symposium on*, pages 32–46, May 2005.